Redes de Ordenadores para Robots y Máquinas Inteligentes
Práctica 3: NAT y Cortafuegos *(firewalls)*

GSyC
Departamento de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación
URJC

Febrero de 2024

Antes de comenzar, descarga tu escenario del siguiente enlace donde deberás introducir tu número de DNI (8 dígitos) con la letra correspondiente:

`https://mobiquo.gsyc.urjc.es/practicas/ror/p3.html`

En la figura 1 se representa un conjunto de subredes y máquinas (`pc1`, `pc2`, `pc4`, `pc5`, `r1`, `r2` y `firewall`) que pertenecen a una determinada empresa y su conexión a Internet a través de la máquina `firewall`. La empresa tiene definidas un conjunto de subredes de ámbito privado:

- 10.X.0.0/24: `r1(eth1)`, `pc1`, `pc2`
- 10.X.1.0/24: `firewall(eth0)`, `r1(eth0)`, `r2(eth0)`
- 10.X.2.0/24: `r2(eth1)`, `pc3`

Adicionalmente, la empresa tiene las máquinas `pc4` y `pc5` que se encuentran en una subred pública: 100.X.0.0/24. Estas máquinas proporcionan servicios básicos de la empresa: servidor de HTTP y servidor de fecha y hora. A esta zona de la red interna donde la empresa tiene una o varias subredes públicas para ofrecer servicios a Internet se le denomina zona desmilitarizada o DMZ (DeMilitarized Zone).

Todas las máquinas de la empresa se conectan a Internet a través de la máquina `firewall` y la subred 100.X.1.0/24.

En este escenario, se considera que Internet está formado por las siguientes máquinas: `r3`, `r4`, `r5`, `pc6` y `pc7` que se encuentran conectadas a las siguientes subredes públicas:

- 100.X.1.0/24: `r3(eth0)`
- 100.X.2.0/24: `r3(eth1)`, `r5(eth2)`
- 100.X.3.0/24: `r3(eth2)`, `r4(eth2)`
- 100.X.4.0/24: `r4(eth1)`, `r5(eth0)`
- 100.X.5.0/24: `r4(eth0)`, `pc6`
- 100.X.6.0/24: `r5(eth1)`, `pc7`

Arranca de una en una todas las máquinas de la figura.

1. Introducción

A continuación se proporcionan algunos consejos para facilitar la realización de la práctica.

1.1. Edición y ejecución de *scripts*

En esta práctica se configurará la máquina `firewall` para que actúe como traductor de direcciones y como cortafuegos. Habrá que definir varias reglas utilizando `iptables`. Por este motivo, es recomendable guardar dichas reglas en un fichero *script de shell*.

Considera la posibilidad de editar y guardar el script en el sistema de ficheros de la máquina real, ejecutándolo desde dentro de la máquina virtual. Así, si tu script `fw.sh` está almacenado directamente en tu HOME de la máquina real, podrías editarlo en ella con un editor gráfico (por ejemplo, `gedit`) y luego ejecutarlo en la máquina `firewall` escribiendo dentro de esa máquina virtual:

```
/hosthome/fw.sh
```

1.2. Comprobación de la configuración del *firewall*

Durante la práctica frecuentemente tendrás que ir comprobando que el *firewall* está correctamente configurado, es decir:

- deja pasar el tráfico que está permitido.
- impide el paso del tráfico que debe ser bloqueado.
- realiza la traducción de direcciones IP necesaria para que no aparezcan en Internet paquetes con direcciones privadas

Para ello deberás emplear la herramienta *netcat* (`nc`) (ya utilizada en prácticas del curso pasado) que permite arrancar aplicaciones TCP y UDP en modo cliente o servidor.

El enunciado de la práctica te irá indicando cuándo y en qué máquinas debes lanzar un cliente o un servidor TCP o UDP para ir probando la configuración del *firewall*. Consulta la documentación adjunta para recordar la sintaxis de *netcat*.

2. Traducción de direcciones y puertos en el *firewall*: tabla `nat`

2.1. Clientes en la red privada, servidores externos

Configura un *script* `fw1.sh` en el *firewall* para que:

- se borren las reglas que hubiera configuradas previamente en la tabla `nat`
- se reinicien los contadores de la tabla `nat`
- se realice la traducción de direcciones para el tráfico saliente de las redes privadas (SNAT) y su correspondiente tráfico de respuesta.

Incluye el script en la memoria.

2.1.1. Pruebas con TCP

Ejecuta el *script* `fw1.sh` de 2.1.

1. Captura el tráfico en `r3-eth0` (iptables-01.cap) y en `firewall-eth0` (iptables-02.cap) para ver los paquetes dentro de la red de la Empresa y por Internet. Arranca las siguientes aplicaciones:

   - `nc` como servidor TCP en `pc6`, puerto 7777
   - `nc` como cliente TCP en `pc1`

   Sin escribir nada ni en el cliente ni en el servidor, consulta la información de `ip_conntrack` del `firewall` cada medio segundo. Para hacerlo automáticamente, en vez de repetir el comando utiliza `watch` de la siguiente forma:

   ```
   firewall:~# watch -n 0.5 cat /proc/net/ip_conntrack
   ```

   Explica el número de paquetes que se han observado en cada sentido, razonando la respuesta, indicando de qué paquetes se trata (recuerda que estamos ante una conexión TCP).

2. Introduce una palabra en la entrada estándar de `pc1`, pulsa `<Enter>` y observa los cambios en `ip_conntrack`. Explica a qué se deben.

3. Realiza un Ctrl+C en el terminal de `pc1` para interrumpir la ejecución de `nc`. Observa los cambios en `ip_conntrack` y explica a qué se deben.

4. Interrumpe las capturas, y estúdialas. En particular, identifica los mismos paquetes en las 2 capturas, y observa cómo cambian las direcciones IP de los mismos paquetes según viajen dentro de la EMPRESA o por INTERNET. Explica el resultado.

5. Consulta la lista de reglas en el `firewall` con:

   ```
   firewall:~# iptables -t nat -L -v -n
   ```

   Obseva qué regla(s) están cumpliendo los paquetes y cuántas veces se cumple(n).

6. Vuelve a repetir la misma prueba anterior (sin necesidad de realizar las capturas de tráfico): lanza servidor y cliente, intercambia tráfico, y termina la conexión. Vuelve a mirar qué regla(s) se están cumpliendo y **cuántas veces** se cumple(n).

2.1.2. Pruebas con UDP

Ejecuta el *script* `fw1.sh` de 2.1 para que se reinicien los contadores de paquetes de iptables, compruébalo consultando la lista de reglas del firewall.

1. Captura el tráfico en `r3-eth0` (iptables-03.cap) y en `firewall-eth0` (iptables-04.cap) para ver los paquetes dentro de la red de la Empresa y por Internet. Arranca las siguientes aplicaciones:

   - `nc` como servidor UDP en `pc6`, puerto 7777
   - `nc` como cliente UDP en `pc2`

   Realiza las siguientes pruebas:

a) Sin escribir nada ni en el cliente ni en el servidor, consulta la información de `ip_conntrack` del `firewall` cada medio segundo. Recuerda que el tráfico es ahora UDP y no hay conexiones propiamente dichas. Explica el resultado.

b) Escribe 5 líneas en el terminal de `pc2` para que se las envíe a `pc6`. Explica el número de paquetes enviados en la información que muestra `ip_conntrack`.

c) Escribe una línea en `pc6` para que se la envíe a `pc2`. Explica nuevamente el número de paquetes en `ip_conntrack`.

d) Observa el poco tiempo que se mantiene la "asociación" entre cliente y servidor en `ip_contrack`. Indica cuánto ha sido.

e) Interrumpe la captura y las ejecuciones de `nc`, explica la captura y cómo ésta se relaciona con la información que has visto en `ip_conntrack`.

2. Consulta la lista de reglas en el `firewall`, e indica cuáles se están cumpliendo y cuántas veces se cumplen.

3. Interrumpe la ejecución de cliente y servidor e inicia una nueva comunicación entre un nuevo cliente y un servidor UDP e intercambia tráfico entre ellos para ver cómo evolucionan las cuentas en la lista de reglas. Explica qué reglas se están cumpliendo ahora y **cuántas veces** se cumplen.

4. Captura de nuevo el tráfico en `r3-eth0` (`iptables-05.cap`) y en `firewall-eth0` (`iptables-06.cap`) para ver los paquetes dentro de la red de la Empresa y por Internet cuando tienes varios clientes desde un mismo puerto origen conectándose a un mismo servidor, para ello inicia:

   - `nc` como servidor UDP en `pc7`, puerto 7777
   - `nc` como cliente UDP en `pc1`, puerto 6666
   - `nc` como cliente UDP en `pc2`, puerto 6666

   Ahora, envía una línea desde `pc1` y después una línea desde `pc2`. Ten en cuenta que `nc` no funciona como las aplicaciones servidoras que pueden atender a varios clientes a la vez. La aplicación `nc` no está preparada para que un servidor se pueda comunicar a la vez con dos clientes, por ello el envío desde `pc2` provocará que `pc7` envíe un ICMP de error a `pc2`. Pero para lo que queremos comprobar este error no es importante, sólo queremos analizar lo que ocurre en el `firewall` con la traducción de direcciones IP y puertos.

   Interrumpe las capturas y analízalas fijándote en las direcciones IP **y puertos** que se utilizan en la red de la EMPRESA y en INTERNET.

2.1.3. Pruebas con ICMP

Ejecuta el *script* `fw1.sh` de 2.1 para que se reinicien los contadores de paquetes de iptables. Vamos a generar tráfico ICMP.

1. Realiza una captura en `pc6` (`iptables-07.cap`)) y otra en `r1(eth1)` (`iptables-08.cap`).

2. Ejecuta el siguiente comando en `pc1` (recuerda sustituir la X por el número que te corresponde):

   ```
   pc1:~# ping -c 2 100.X.5.60
   ```

3. Interrumpe las capturas anteriores.

4. Consulta la información de `ip_conntrack` del `firewall`. Verás que no aparece nada. Recuerda que esto se debe a que las "conexiones" que se consideran para los paquetes ICMP es una diferente entre cada *echo request* y su correspondiente *echo reply*, asociación que se "olvida" justo después del *echo reply*.

5. Consulta la lista de reglas en el `firewall`, y mira cuáles se están cumpliendo y **cuántas veces**, relaciona esta información con los mensajes capturados.

2.2. Servidores en la red privada, clientes externos

Aunque en una red como la que aparece en la figura, lo habitual es colocar los servidores accesibles desde el exterior en la zona DMZ, para ver cómo funciona DNAT, vamos a permitir que haya servidores accesibles desde el exterior en la red privada interna.

2.2.1. Apertura de puertos TCP

Realiza un nuevo *script* `fw2.sh` en el *firewall* para que:

- se borren las reglas que hubiera configuradas previamente en la tabla `nat`
- se reinicien los contadores de la tabla `nat`
- el tráfico de entrada al firewall destinado al puerto TCP 80 sea redirigido a `pc3`, puerto 80.

Incluye el script en la memoria. Ejecuta dicho script y arranca las siguientes aplicaciones:

- Inicia una captura en `r2(eth1)`(`iptables-09.cap`) y en `r4(eth0)` (`iptables-10.cap`).
- `nc` como servidor TCP en `pc3`, puerto 80
- `nc` como cliente TCP en `pc6`, de forma que su tráfico lo reciba el servidor de `pc3` (NOTA: presta especial atención a los parámetros con los que debes lanzar este cliente). Indica en la memoria el comando que has usado para lanzar el cliente y explica por qué lo has hecho así.
- Escribe una palabra en el lado cliente y pulsa `<Enter>`
- Interrumpe la ejecución del cliente y el servidor.
- Interrumpe las capturas.

Explica los siguientes resultados:

1. El resultado observado en `ip_conntrack` y la traducción de direcciones IP y puertos realizada.
2. La lista de reglas en el `firewall`, indica cuáles se están cumpliendo y cuántas veces.
3. Relaciona las reglas que se han cumplido con los datos de los mensajes capturados en los ficheros.

2.2.2. Apertura de puertos UDP

Modifica el *script* `fw2.sh` para que, adicionalmente:

- el tráfico de entrada al firewall destinado al puerto UDP 5001 sea redirigido a `pc1`, puerto 5001
- El tráfico de entrada al firewall destinado al puerto UDP 5002 sea redirigido a `pc2`, puerto 5001

Incluye el script en la memoria. Ejecuta el script que acabas de modificar y arranca las aplicaciones:

- Inicia una captura en `r1(eth1)`(`iptables-11.cap`), en `r4(eth0)` (`iptables-12.cap`) y en `r5(eth1)` (`iptables-13.cap`).
- `nc` como servidor UDP en `pc1`, puerto 5001

- `nc` como servidor UDP en `pc2`, puerto 5001
- `nc` como cliente UDP en `pc6`, de forma que su tráfico lo reciba el servidor de `pc1`. Indica el comando que has utilizado para lanzar el cliente y explica por qué.
- `nc` como cliente UDP en `pc7`, de forma que su tráfico lo reciba el servidor de `pc2`. Indica el comando que has utilizado para lanzar el cliente y explica por qué.
- Escribe una palabra en el lado cliente de `pc6` y `pc7` y pulsa `<Enter>`
- Interrumpe la ejecución de los clientes y servidores.
- Interrumpe las capturas.

Explica los siguientes resultados:

1. El resultado observado en `ip_conntrack` y la traducción de direcciones IP y puertos realizada.
2. Consulta la lista de reglas en el `firewall` e indica cuáles se están cumpliendo y cuántas veces.
3. Relaciona las reglas que se han cumplido con los datos de los mensajes capturados en los ficheros.

3. Filtrado de tráfico en el *firewall*: tabla `filter`

Crea un *script* `fw3.sh` en el `firewall` partiendo de la configuración de traducción de direcciones realizada en `fw1.sh` (clientes en la red privada, servidores externos) al que se le añada la siguiente configuración (todas en el mismo *script*). Descripción de las **especificaciones**:

1. Reiniciar la tabla `filter`: borrar su contenido y reiniciar sus contadores.
2. Fijar las políticas por defecto de las cadenas de la tabla `filter`, haciendo que por defecto se descarte todo el tráfico en el `firewall` excepto los paquetes que cree el propio `firewall` (configuración habitual en un *firewall*).
3. Permitir el tráfico de entrada dirigido a las aplicaciones que se están ejecutando en el propio `firewall` únicamente si este tráfico tiene su origen en las subredes privadas de la empresa.
4. Permitir todo el tráfico saliente desde las subredes privadas hacia Internet y el tráfico de respuesta al saliente.

   Ten en cuenta que como has partido del *script* `fw1.sh`, en dicho *script* ya tenías las reglas de la tabla `nat` para la traducción de la dirección IP de origen de los paquetes que reenvía el `firewall` y los paquetes del tráfico entrante de respuesta a éste.
5. Permitir desde Internet únicamente el tráfico entrante nuevo hacia la zona DMZ según las siguientes reglas:
   - acceso a un servidor *echo* existente en `pc4` (UDP, puerto 7). El servidor de *echo* es un servidor que al enviarle una cadena de caracteres, devuelve la misma cadena que se le ha enviado. Para comprobar el acceso a este servidor utiliza `nc` como cliente desde otra máquina.
   - acceso a un servidor *daytime* existente en `pc5` (UDP, puerto 13). El servidor *daytime* es un servidor que al enviarle algo, devuelve la fecha y hora de la máquina donde está instalado. Para comprobar el acceso a este servidor utiliza `nc` como cliente desde otra máquina.

Para este tipo de tráfico configura además regla/s con **acción LOG** para que cada vez que se permita el tráfico UDP descrito anteriormente, se deje un mensaje en el fichero de LOG del sistema.

6. Permitir únicamente la comunicación entre la red privada y la zona DMZ de la siguiente forma:

   - acceso desde `pc1` a un servidor de *echo* (TCP, puerto 7) existente en `pc4`. En este caso, como todas las máquinas involucradas en la comunicación pertenecen al ámbito privado de la empresa, no es necesario que realices traducción de direcciones. Para este tipo de tráfico configura además regla/s con **acción LOG** para que cada vez que se permita este tráfico TCP, se deje un mensaje en el fichero de LOG del sistema.

7. Desde la zona DMZ NO se debe permitir iniciar ninguna comunicación con la red privada ni con el propio `firewall`.

Incluye el script en la memoria.

Prueba el resultado de la configuración del script:

1. Sólo aplicaciones de la red privada pueden comunicarse con aplicaciones ejecutándose en el `firewall`:

   - Ejecuta el script de configuración para que se reinicien los contadores.
   - Prueba a lanzar un servidor TCP en el puerto 1111 de la máquina firewall. Lanza un cliente en `pc1` que se comunique con ese servidor. Indica qué reglas del firewall se están cumpliendo.
   - Lanza un cliente en `pc6` que trate de comunicarse con ese servidor. Indica qué provoca que ese tráfico sea descartado.

2. Comunicación desde un cliente de la red privada con un servidor cualquiera en el exterior:

   - Ejecuta el script de configuración para que se reinicien los contadores.
   - Prueba a lanzar un servidor TCP en pc6 en el puerto 1111. Realiza una captura de tráfico en `r3(eth0)` (`iptables-14.cap`). Prueba a lanzar un cliente con `nc` desde `pc1` para que se comunique con este servidor.
   - Consulta la lista de reglas en el `firewall` e indica cuáles se están cumpliendo y cuántas veces se cumplen como resultado de esta comunicación, tanto en la tabla nat como en la tabla filter.

3. Comunicación entre `pc7` y servidor de UDP en el puerto 7 de `pc4`:

   - Ejecuta el script de configuración para que se reinicien los contadores.
   - En el escenario se encuentra lanzado en `pc4` un servidor UDP en el puerto 7 (echo). Realiza una captura de tráfico en `pc4(eth0)` (`iptables-15.cap`). Prueba a lanzar un cliente con `nc` desde `pc7` para que se comunique con este servidor.
   - Consulta la lista de reglas en el `firewall` e indica cuáles se están cumpliendo y cuántas veces se cumplen como resultado de esta comunicación.
   - Incluye en la memoria los mensajes que esta comunicación ha provocado en el fichero de log.

4. Comunicación entre `pc6` y servidor de UDP en el puerto 13 de `pc5`:

   - Ejecuta el script de configuración para que se reinicien los contadores.

- En el escenario se encuentra lanzado en `pc5` un servidor UDP en el puerto 13 (daytime). Realiza una captura de tráfico en `pc5(eth0)` (`iptables-16.cap`). Prueba a lanzar un cliente con `nc` desde `pc6` para que se comunique con este servidor.
- Consulta la lista de reglas en el `firewall` e indica cuáles se están cumpliendo y cuántas veces se cumplen como resultado de esta comunicación.
- Incluye en la memoria los mensajes que esta comunicación ha provocado en el fichero de log.

5. Comunicación entre `pc1` y servidor de TCP en el puerto 7 de `pc4`:
   - Ejecuta el script de configuración para que se reinicien los contadores.
   - En el escenario se encuentra lanzado en `pc4` un servidor TCP en el puerto 7 (echo). Realiza una captura de tráfico en `pc4(eth0)` (`iptables-17.cap`). Prueba a lanzar un cliente con `nc` desde `pc1` para que se conecte con este servidor.
   - Consulta la lista de reglas en el `firewall` e indica cuáles se están cumpliendo y cuántas veces se cumplen como resultado de esta comunicación. Es importante que observes que las reglas de la tabla `filter`, si se cumple la condición, se aplican con cada paquete que atraviesa el `firewall` y este comportamiento es diferente a lo que ocurría con las reglas de la tabla `nat`.
   - Incluye en la memoria los mensajes que esta comunicación ha provocado en el fichero de log.

Entrega de la práctica

Es necesario entregar la siguiente documentación en un fichero `p1.zip` o `p1.tgz`:

- Memoria donde se explique razonadamente el diseño y la configuración de cada uno de los apartados de este enunciado.
- Capturas de tráfico desde `iptables-01.cap` a `iptables-17.cap`.

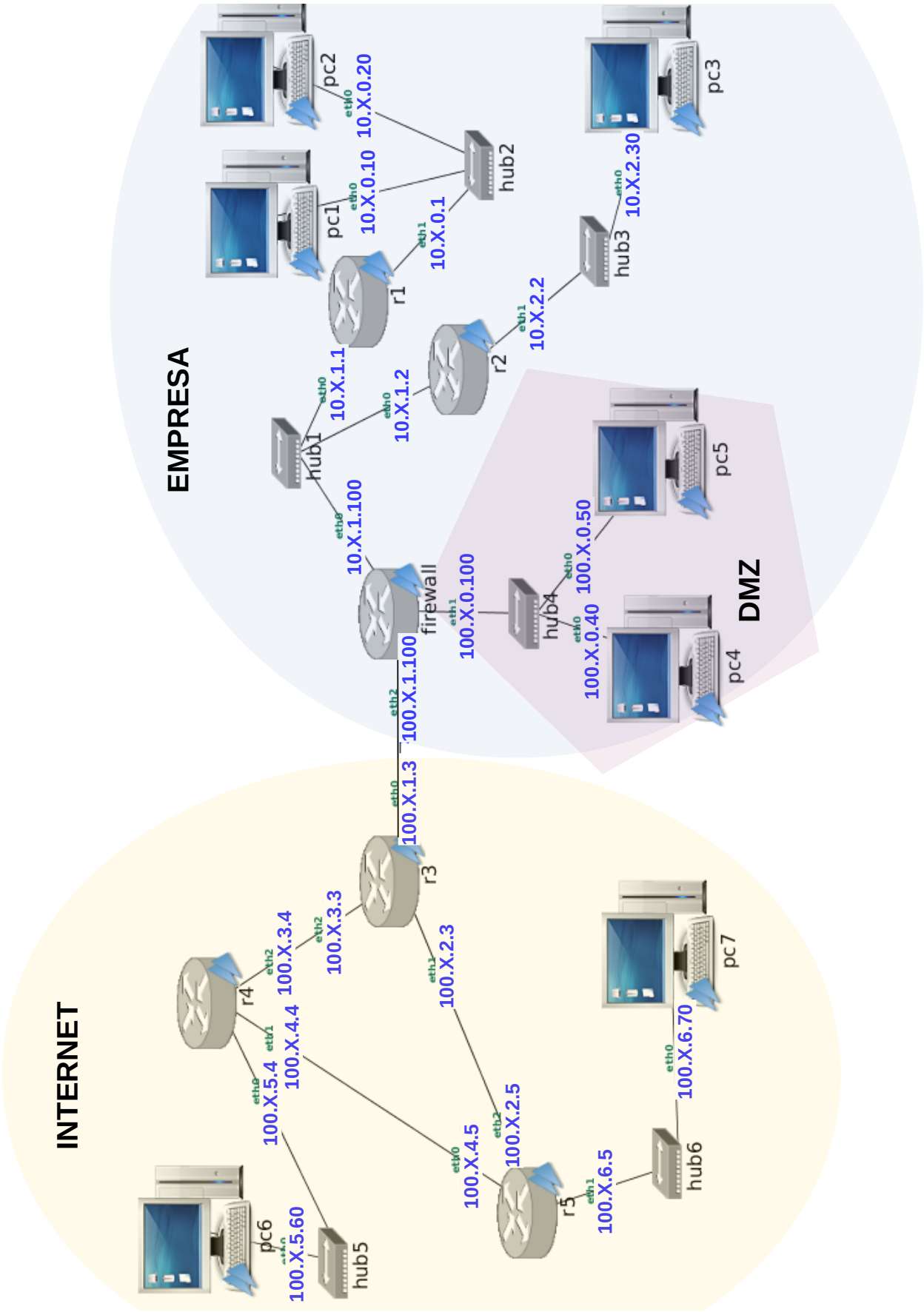

Figura 1: Escenario de red para la configuración de un firewall